

Lập trình mạng (Network Programming)

Chương 6. Hyper Text Transfer Protocol
(HTTP)

HTTP Request và Response

- Client giao tiếp với server. Bước 1: Get request



- Client giao tiếp với server. Bước 2: HTTP response



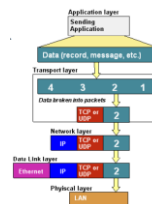
Lớp Uri

- Biểu diễn/định danh một tài nguyên trên mạng (intranet/internet)
- Là một địa chỉ trên trình duyệt ánh xạ đến tài nguyên trên server
- Lớp Uri định nghĩa các phương thức và thuộc tính cho việc xử lý các URI (tách, so sánh, kết hợp)
- Các thuộc tính của lớp Uri là chỉ đọc
- Ví dụ: tạo thể hiện Uri với URI: 'http://mytest.com/test.html/'

```
Uri myUri = new Uri("http://myweb.com/test.html");
Stream strm = wc.OpenRead(myUri);
StreamReader sr = new StreamReader(strm);
String response = sr.ReadLine();
```

Giao thức HTTP

- HTTP: một trong các giao thức chuẩn về mạng internet – trao đổi thông tin giữa client và server
- Lớp WebClient (namespace System.Net): trao đổi thông tin qua HTTP
- Các lớp WebRequest vàWebResponse: xây dựng các ứng dụng web
- Web Services trao đổi thông tin trực tiếp với web server để gửi và nhận thông tin



Lớp WebClient

- Lớp WebClient cung cấp các phương thức để gửi và nhận dữ liệu từ mạng cục bộ, intranet hoặc internet (xác định bởi URI)
- 3 phương thức lấy thông tin từ web server
 - DownloadData(): download dữ liệu về mảng byte từ một URI
 - DownloadFile(): download dữ liệu về file từ một URI
 - OpenRead(): mở một stream chỉ đọc để download dữ liệu từ URI
- Các phương thức yêu cầu tham số URI để xác định vị trí website. URI phải theo chuẩn HTTP URI

Phương thức DownloadData()

- Lấy các thẻ HTML thô từ HTTP request và đặt trong mảng byte
- Khi message lấy về mảng byte, ta có thể xử lý HTML → tìm từ khóa và các giá trị trong trang web
- Ví dụ: download trang web và hiển thị

```
1 using System;
2 using System.Net;
3 using System.Text;
4 class DownloadDataTest
5 {
6     public static void Main(string[] args)
7     {
8         WebClient wc = new WebClient();
9         byte[] response = wc.DownloadData(args[0]);
10        Console.WriteLine(Encoding.ASCII.GetString(response));
11        Console.ReadKey();
12    }
13 }
```

```
C:\>DownloadDataTest http://localhost/test.htm
<html>
<title>This is a test web page</title>
<body>
<h1>This is a test web page</h1>
</body>
</html>
C:\ >
```


Phương thức UploadData

- Upload dữ liệu từ một mảng byte tới vị trí do URI xác định
 - UploadData(string URI, byte[] array);
 - UploadData(string URI, string method, byte[] array);
- Phương thức UploadData ngầm định sử dụng phương thức HTTP POST để chuyển dữ liệu tới web server

• Ví dụ

```
public static void Main(string[] argv)
{
    WebClient wc = new WebClient();
    string data = "This is the data to post";
    byte[] dataarray = Encoding.ASCII.GetBytes(data);
    wc.UploadData(argv[0], dataarray);
}
```

Phương thức UploadFile

- Upload file lên web server
 - UploadFile(string URI, string filename);
 - UploadFile(string URI, string method, string filename);
- Chú ý: phải có ít nhất 2 tham số tên file và địa chỉ URI

• Ví dụ

```
6 public static void Main(string[] argv)
7 {
8     WebClient wc = new WebClient();
9     wc.UploadFile(argv[0], argv[1]);
10 }
```

Phương thức UploadValues

- Không upload dữ liệu, upload cặp tên/giá trị lên web server
 - UploadValues(String uri, NameValueCollection nvc);
 - UploadValues(Uri uri, NameValueCollection nvc);
- Phương thức UploadValues sử dụng đối tượng NameValueCollection để tạo cặp tên và giá trị


```
NameValueCollection nvc = new NameValueCollection();
nvc.Add("lastname", "Abc");
nvc.Add("firstname", "Def");
byte[] response = wc.UploadValues(uri, nvc);
```
- Để test UploadValues, ta phải có website chấp nhận dữ liệu HTML form

Các lớp WebRequest và WebResponse

- Thuộc System.Net, cung cấp các chức năng nâng cao cho lập trình mạng
- Là các lớp trừu tượng (abstract) → không sử dụng trực tiếp như là các đối tượng trong chương trình
 - Phụ thuộc vào tiền tố (prefix) URI xác định trong tham số URI
- Với các URI http và https, sử dụng các lớp HttpWebRequest và HttpResponse để xử lý giao tiếp HTTP với web server
- Với file URI (truy cập các trang HTML cục bộ), sử dụng các lớp FileWebRequest và FileWebResponse

Lớp HttpWebRequest

- Cung cấp các chức năng nâng cao để lấy trang web sử dụng HTTP từ web server
- Để tạo đối tượng HttpWebRequest, ta phải chuyển kiểu kết quả của phương thức WebRequest.Create


```
HttpWebRequest hwr = (HttpWebRequest)WebRequest.Create("http://remotehost/webpage.htm");
```
- Sau khi đối tượng HttpWebRequest được tạo, ta có thể đặt các trường header HTTP bằng các thuộc tính của lớp HttpWebRequest

Web Proxy

- Web proxy server dùng để gửi dữ liệu qua firewall
- Bất cứ web request nào từ mạng cục bộ cũng phải gửi tới web proxy server rồi mới chuyển đến địa chỉ thích hợp trên internet
- Lớp HttpWebRequest cho phép xác định web proxy server để sử dụng cho tất cả các giao tiếp HTTP
- Thuộc tính Proxy định nghĩa URI của web proxy server
- Lớp WebProxy phải được sử dụng để xác định giá trị cho thuộc tính Proxy

```
HttpWebRequest hwr = (HttpWebRequest)WebRequest.Create("http://remotel1.ispnet2.net);
WebProxy proxySrv = new WebProxy("http://proxy1.ispnet.net:2000");
hwr.Proxy = proxySrv;
```

Sau khi định nghĩa web proxy server, ta có thể gửi/nhận các HTTP message

Gửi dữ liệu

- Sử dụng phương thức `GetRequestStream()`

```
HttpRequest hwr = (HttpRequest)WebRequest.Create("http://localhost");
Stream strm = hwr.GetRequestStream();
StreamWriter sw = new StreamWriter(strm);
sw.WriteLine(data);
```

- Phương thức này trả về đối tượng `Stream` có thể gửi dữ liệu trực tiếp lên web server
- Đối tượng `Stream` được trả về qua phương thức `GetRequestStream()` – chỉ dùng để gửi dữ liệu, không lấy dữ liệu

Request dữ liệu

- Hai cách để lấy response từ đối tượng `HttpRequest`
 - `GetResponse()`: đặt response từ web server trong đối tượng `HttpWebResponse`
 - `BeginGetResponse`: sử dụng hàm không đồng bộ để đặt response từ web server trong đối tượng `HttpWebResponse`
- Cả hai phương thức trả về response của server như đối tượng `HttpWebResponse`

Lớp HttpWebResponse

- Là lớp dẫn xuất từ lớp `WebResponse` và được tạo ra để chuyên xử lý HTTP response từ web server
- Gồm các phương thức và thuộc tính để trích xuất thông tin từ trang web nhận được
- Lớp `HttpWebResponse` chứa các thuộc tính cho hầu hết các trường của HTTP header

Lớp HttpWebResponse

- Các thuộc tính

Property	Gets
<code>CharacterSet</code>	The character set used in the response
<code>ContentEncoding</code>	The encoding method used in the response
<code>ContentLength</code>	The length of the content returned in the response
<code>ContentType</code>	The content type of the response
<code>Cookie</code>	Cookie with the cookies associated with the response
<code>Headers</code>	The HTTP headers in the response
<code>LastModified</code>	The date and time the contents of the response were modified
<code>Method</code>	The HTTP method used in the response
<code>ProtocolVersion</code>	The HTTP version used for the response
<code>ResponseUri</code>	The URI of the server that responded to the request
<code>Secure</code>	The name of the server that sent the response
<code>StatusCode</code>	The HTTP status code of the response
<code>StatusDescription</code>	The HTTP text status phrase used in the response

Lấy các web response header

- Có hai cách để lấy header
 - Sử dụng các thuộc tính trường header để trích xuất thông tin header cho từng trường header
 - Sử dụng thuộc tính header để tạo một đối tượng `WebHeaderCollection`. Đối tượng này chứa tất cả các header trong message HTTP response

```
HttpRequest hwr = (HttpRequest)WebRequest.Create(uri);
HttpWebResponse hwrsp = (HttpWebResponse)hwr.GetResponse();
WebHeaderCollection whc = hwrsp.Headers;
for (int i = 0; i < whc.Count; i++) {
    headers.Items.Add(whc.GetKey(i) + " = " + whc.Get(i));
}
```

Lấy dữ liệu HTML từ Web response

- Đối tượng `HttpRequest` sử dụng phương thức `GetResponse` để trả về một đối tượng `HttpWebResponse`
- Để lấy thông tin từ HTTP response, sử dụng phương thức `GetResponseStream` để trả về một đối tượng `Stream` có thể đọc dữ liệu từ thông báo HTTP response
- Sau khi đối tượng `Stream` được tạo, ta có thể tạo đối tượng `StreamReader` để lấy từng dòng của HTML code trong thông báo HTTP response

```
HttpRequest hwr = (HttpRequest)WebRequest.Create(uri);
HttpWebResponse hwrsp = (HttpWebResponse)hwr.GetResponse();
Stream strm = hwrsp.GetResponseStream();
StreamReader sr = new StreamReader(strm);
response.Items.Add(sr.ReadLine());
```

Sử dụng Cookies

- Thuộc tính Cookie cho phép nhìn thấy tất cả các cookies mà web server gửi trong response message
- Cookie được dùng để lưu trữ thông tin từ web server trên client
- Có thể đọc từ web server để xác định và theo dõi thông tin trạng thái của client

```
HttpWebRequest hwr = (HttpWebRequest)WebRequest.Create("http://www.amazon.com/");
hwr.CookieContainer = new CookieContainer(); // empty cookie
HttpWebResponse hwrsp = (HttpWebResponse)hwr.GetResponse();
hwrsp.Cookies = hwr.CookieContainer.GetCookies(hwr.RequestUri);
foreach(Cookie cky in hwrsp.Cookies) {
    Console.WriteLine(cky.Name + " = " + cky.Value);
}
```

Web Services

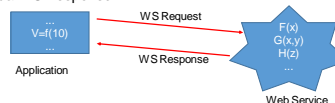
- Cho phép tích hợp qua mạng giữa các ứng dụng được viết bằng những ngôn ngữ khác nhau trên các nền khác nhau
 - Tích hợp trong cùng host (cùng platform, cùng ngôn ngữ)
 - Bao gồm mã nguồn
 - Thư viện liên kết tĩnh/động (lib, dll)
 - Tích hợp qua mạng
 - Remote Procedure Call
 - Web Service
- Dữ liệu truyền qua web service sử dụng SOAP (Simple Object Access Protocol) – tận dụng chuẩn kết nối HTTP cùng với XML (eXtensible Markup Language) để định dạng dữ liệu truyền

Web Services Diagram

- Web service server
 - Cung cấp các phương thức ứng dụng qua IIS web server (ASP.NET)
 - Các phương thức nhận dữ liệu từ remote client sử dụng SOAP, xử lý dữ liệu và trả kết quả
- Đối tượng Proxy
 - Cho phép remote client chuyển dữ liệu tới web service server
- Client application
 - Sử dụng phương thức web service trên server
 - Có thể là ứng dụng cục bộ để lấy thông tin trên server

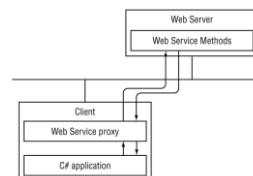
Web Services

- Là thành phần phần mềm cung cấp các phương thức có thể gọi qua mạng
 - Tài liệu WSDL (Web Services Description Language) xác định các phương thức mà Web Service cung cấp
 - Gọi phương thức của Web Service: gửi WS Request và nhận kết quả qua WS Response



Diagram

- Tiến trình



Tạo web service server

- Cài đặt IIS web server
- Tạo web service: xây dựng các phương thức chạy trên IIS server (kết hợp các phương thức trên ASP.NET)
- Trang web service có phần mở rộng là .asmx
- Khai báo
 - <%@ WebService Language="c#" Class="Tên lớp"%>
 - Khai báo các lớp cho web service, mỗi phương thức có chỉ dẫn [WebMethod] để định danh đó là phương thức web service